DaaS: Dew Computing as a Service for Intelligent Intrusion Detection in Edge-of-Things Ecosystem

Parminder Singh, *Member, IEEE,* Avinash Kaur, Gagangeet Singh Aujla, *Senior Member, IEEE*, Ranbir Singh Batth, *Member, IEEE,* and Salil Kanhere, *Senior Member, IEEE*

***Abstract*—Edge-of-Things (EoT) enables the seamless transfer of services, storage and data processing from the Cloud layer to Edge devices in a large-scale distributed Internet of Things (IoT) ecosystems (e.g., Industrial systems). This transition raises the privacy and security concerns in the EoT paradigm distributed at different layers. Intrusion detection systems are implemented in EoT ecosystems to protect the underlying resources from attackers. However, the current intrusion detection systems are not intelligent enough to control the false alarms, which significantly lower the reliability and add to the analysis burden on the intrusion detection systems. In this article, we present a DaaS, Dew Computing as a Service for intelligent intrusion detection in EoT ecosystems. In DaaS, a deep learning-based classifier is used to design an intelligent alarm filtration mechanism. In this mechanism, the filtration accuracy is improved (or sustained) by using deep belief networks. In the past, the cloud-based techniques have been applied for offloading the EoT tasks, which increases the middle layer burden and raises the communication delay. Here, we introduce the dew computing features which are used to design the smart false alarm reduction system. DaaS, when experimented in a simulated environment, reflects lower response time to process the data in the EoT ecosystem. The revamped DBN model achieved the classification accuracy up to 95%. Moreover, it depicts a 60% improvement in the latency and 35% workload reduction of the cloud servers as compared to Edge intrusion detection system.**

***Index Terms*—Intrusion detection, Edge-of-Things, Dew Computing, Smart false alarm filter, Deep belief networks.**

I. Introduction

In the recent era, adoption of solutions of Internet-of-Things (IoT) and new end-user applications high-performance demands, makes it difficult for the cloud model to provide solutions to new needs. The strong networked and remote computing resources as storage and computing resources are offered by the cloud [1]. The cloud aims to guarantee the low utilization of network bandwidth for virtual resources migration, vital and versatile IoT devices productivity during the movement of information to the cloud. Hence, providing a solution to these problems, the Edge computing framework evolved as a compliment to the Cloud systems and IoT networks [2]. This new technology known as Edge-of-Things (EoT) computing, deals with the processing of information and also the movement of administration supply from Cloud to nearby Edge devices. The EoT enables stockpiling of information, registering it and processing it in near real-time. Thus, is powerful over administration and utilization as stream handling of ongoing video, gaming, etc. EoT provides the seamless transfer of services, storage and data processing from the Cloud layer to Edge devices in a large-scale distributed Internet of Things ecosystems (eg., Industrial systems). This transition raises the privacy and security concerns in the EoT paradigm distributed at different layers. Intrusion detection systems are implemented in EoT ecosystems to protect the underlying resources from attackers.

Anomaly-based Intrusion Detection Systems (IDSs) have a series of concerns with false alarm rates. The generic approach for this issue is to limit or reduce the false alarm rate by monitoring the incoming traffic and finding malicious events from abnormal network/system behavior [3]. IDS tries to find the data which deviates from the model up to a certain threshold, known as *anomalies or outliers*. Normality in the traffic is different across the system, and need a careful design of IDS with accurate data representation. The accuracy of IDS can be represented with the probability of positive detection of anomalies. As far as the application domain, the effectiveness of IDS is observed from a minimum False Positive Rate (FPR), instead of correct identification of anomalies known as the true positive rate (TPR). In a real-world implementation, the anomaly-based IDSs face a large number of false alarms in one day. The FPR is much higher in anomaly-based IDSs as compare to other IDSs, because it is challenging to prepare the significant normal profile for the anomaly-based IDSs in most of the cases. To overcome this problem, an intelligent filter for false alarm can be a key solution. Further, the accuracy of the filter could be enhanced with the adaptive selection of a machine-learning algorithm to minimize the FPR [4].

Intelligent IDS needs to perform some operation on workload in repositories or incoming traffic. Cloud computing provides a solution to this issue by offering the infrastructure for a specific requirement. This helps to increase the performance of distributed IDS configured in the cloud landscape. However, the applications running in edge, fog and IoT environments have a short response time requirement, which may produce a large amount of data [5]. Therefore, cloud computing alone cannot support these applications due to delay in algorithm selection and data processing. Hence, Dew computing has the potential to mitigate these challenges. In the age of cloud computing, dew computing comes with the resolution to use the desktop PC, mobile devices to create a cloud-

P. Singh, A. Kaur, and R. S. Batth are with the School of Computer Science and Engineering, Lovely Professional University, India, 144411. email: parminder.16479@lpu.co.in, avinash.14557@lpu.co.in, ranbir.21123@lpu.co.in

G. S. Aujla is with the Department of Computer Science, Durham University, United Kingdom. email: gagi_aujla82@yahoo.com, gagangeet.s.aujla@durham.ac.uk

S. Kanhere is with the University of New South Wales, Australia. email: salil.kanhere@unsw.edu.au

like environment. The local devices provide rich functionality without cloud computing, but also connected with cloud for the expansion of services. Motivated from this fact, in this article, we proposed an intelligent intrusion detection system for Edge Ecosystem using Dew computing as a Service (DaaS). IDS using dew computing paradigm to perform the intrusion detection near to edge devices. For this purpose, we design, **DaaS**, Dew Computing as a Service for intelligent intrusion detection in EoT ecosystems. The major contributions of DaaS are mentioned below:

- We introduce the new framework to improve the performance of the intrusion detection system using Dew computing. Dew computing has the potential to work as a cloud in the local environment with the collaboration of the public cloud to reduce the communication delay and cloud server workload.
- The Restricted Boltzmann Machine (RBM) model has been improved with sparsity penalty to increase the performance of the DBN model.
- The evaluation of the proposed and existing technique has been conducted on the custom testbed. The experiment results demonstrate that our proposed approach can greatly reduce the cloud server workload and communication delay in the adaptation of algorithms.

A. Article Structure

The rest of this article is organized as follows: Section 2 presents a brief overview and the state of the art analysis. Section 3 presents our proposed DaaS framework with revamped DBN model. Section 4 provides the performance evaluation and experimental results. Finally, section 5 presents the concluding remarks and future directions.

II. Related Work

Various existing proposals have contributed to the above-described challenges in different ways. A few of them are discussed in the subsequent sections.

A. Distributed Intrusion Detection

To enhance the performance of single IDSs, the Collaborative Intrusion Detection Networks (CIDNs) were devised but they have contained very little data about the environment in which the system is deployed. Their performance could be increased by taking useful information from the nodes of the deployed IDS environment. However, inside attacks are the major concern in this scenario. Furthermore, CIDNs can be protected from inside threats with careful design of the trust-based mechanism. Duma et al. [6] developed an overlay IDS based on the P2P mechanism that manages the trust by correlating an adaptive scheme with the trust-aware engine. The low-quality peers and untrusted warnings can be filtered out with the proposed trust-aware correlation engine. The peer's experience is used in adaptive trust management to estimate trustworthiness. Meng et al. [7] designed a Bayesian-based trust mechanism for medical smartphone networks to find the inside attacks. The authors evaluated the proposed model is a realistic scenario.

Fung and Boutaba [8] adopted a challenge-based approach to computing the trustworthiness from challenge answers. In their previous work, the HIDS collaboration framework is proposed, which computes the trustworthiness from forgetting the factor of their own experience. More importance is given to the peer's recent experience in forgetting factors. After that, the Dirichlet-based approach is applied to improve the trustworthiness level in IDS nodes with interactive confront [8]. This proved to be a robust approach that is equipped with a scalability feature to handle common threats. A lot of benefits can reap from CIDNs challenge-based approach. Li et al. [9] proposed a challenge-based CIDNs to gain the performance. It aims to detect the sensitivity of the intrusion on different types of nodes. *Intrusion sensitivity* notion is proposed to calculate the sensitivity of an IDS for detecting various types of intrusions. In the case of signature-based IDS, nodes are considered more powerful, if they contain large no. of signatures or rules. The trust management model [10] [11] developed using a machine learning approach for autonomous assignment of intrusion sensitivity in CIDNs. In [12], a case study has been conducted to measure the effect of intrusion sensitivity in the defence of pollution attacks in a malicious peer-group interacting with each other based on the ranking of FPR. The results demonstrate that intrusion sensitivity is key to emphasize the expert node's impression and efficiently reduce the malicious node's trust.

The advanced attacks are conducted by many researchers because common attacks are automatically defended with the challenge mechanism. Li et al. [12] designed a passive message fingerprint attack (PMFA), a collusion type of attack that acts passively to steal the identity and messages. The results demonstrate that these attacks can maintain trust values while serving normal requests through malicious nodes. After that, they keep serving the normal request to one node and responding abnormally to another one. This happens due to the development of the special case of On-Off attacks (SOOA) [11] . The experimental result indicates that the trust effectiveness of CIDN nodes is interfering with alarm aggregation and trust computation. The machine learning algorithms are used in recent years in network security for anomaly detection, user authentication, malware detection and IP traffic identification. Many new security concerns have been raised due to the machine learning application while outsourcing and distributed learning. This needs the careful deployment of these techniques in the real-world [11].

B. Cloud-based Intrusion Detection

There are many vulnerabilities to be looked upon in cloud [13]. To protect the environment of the cloud from attacks, different intrusion detection systems are investigated. There is a great relationship between intrusion detection systems and cloud computing. =In the cloud environment, for better deployment of cloud, IDS management architecture is introduced that consists of a central management unit and various sensors. Anish *et al.* [14] presented a model to prevent the energy theft in smart grids. The data from power-meter and other sources used to find theft in power network. Combining

VM monitor methods and virtualization technology handles VM based IDs. Grid and Cloud Computing Intrusion Detection System was proposed for the detection of host and network-based attacks. Particularly, every node identifies local events that represent violations of security and interaction with other nodes [15]. Further, for malicious attack identification from different network points Cloud-based Intrusion Detection Service (CBIDS) framework was introduced [16]. In the cloud environment, for addressing security issues, the autonomous agent-based incident detection system is introduced for finding a solution for specific security issues. For enhancement of the security infrastructure of cloud, an Intrusion Detection System as a Service (IDSaaS) is introduced [17]. Hence, it concludes that various computing resources can be offered by the cloud environment to an IDS.

C. Deep Belief Network (DBN) for Intrusion Detection

DBN comes under the category of probability generation model that use the joint probability for the distribution of data samples. Recently, numerous DBN models have been presented. Ding et al. [18] used the opcode sequence to detect the malware. Feed-forward neural network-based DBN is employed for detection and detection performance evaluation. Zhao et al. [19] presented probability-based neural network for DBN. The raw data is converted to low-dimensional data with the help of non-linear capacity of DBN. Even the raw data properties are not affected, but the computational capacity of the model is very high. Kaiser et al. [20] devised a novel convolutions DBN model. However, the drawback of the model is the long duration in the training process. Koo et al. [21] designed supervised DBN model to refine the two-phase technique. This model has the overhead of one more training phase. Tian et al. [22] developed a DBN based IDS to simplify the processing of data. Overfitting and homogeneity issues are mitigated with the help of Gaussian distribution. The model is stacked with various Restricted Boltzmann Machine (RBM) models, which increases the overhead in the overall process.

The performance of DBN is directly proportional to the effectiveness of RBM because the DBN is constructed from the multiple layers of RBM. In the above discussion of DBN depicted that visual layers of RBM are restricted to binary. The primary infusion is to improve the performance of low-level RBM of the DBN. A deterministic algorithm is required to train the RBM layers of RBM to enhance the classification accuracy of the DBN model.

III. DaaS: The Proposed Approach

In this section, we discussed the dew computing and our proposed framework to escalate the classification accuracy and reduce the false alarm rate with dew computing for distributed IDSs with DBN model.

A. Dew Computing

As per Wang [24], the evolution of dew computing is based on two keywords: Collaboration and Independence. Another theory is given by Skala et al. [25], the author defines that

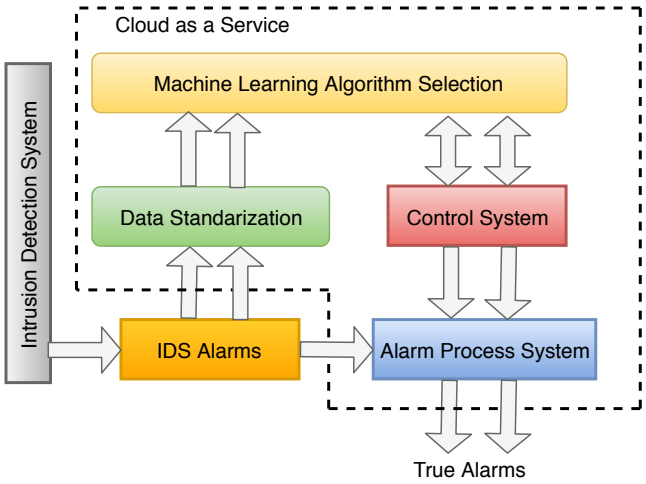

Fig. 1. The high-level architecture of adaptive false alarm reduction scheme using Cloud as a Service [23]

dew computing puts the virtual infrastructure far from the centralized system. Ristov et al. [26] given the "Independence" theory and Y. Wang [24] introduced a new theory known as "Collaboration".

Dew computing is a paradigm for programming which ensures the P2P communication link with the plug-in, ready-to-go, pervasive and ubiquitous services. Dew computing aims to analyze the data from the place of origin with minimal efforts of network management. It is designed for optimal utilization of on-premises computing services such as smart-phones, laptops, and Desktop PCs. The six required characteristics of the dew computing model are Accessibility, Transparency, Re-origination, Scalability, Synchronization, and Rule-based data collection. The model of services provided by dew computing is Infrastructure-as-a-Dew (IaaD), Software-as-a-Dew (SaaD) for products and services. This work makes the system a hierarchical and distributed eco-system. This minimizes the need for cloud computing all the time to the EoT ecosystem.

DaaS is used in the IDS for false alarm detection near to the edge devices. The dew computing is an on-premises cloud-like environment that helps to save energy with scalability features. The edge devices can take the storage/computing services with minimum communication delay.

B. DaaS for Intelligent IDS

The dew computing has the potential to operate on data with better processing, response time and less network load. Distributed IDS could be improved using the intelligent technique for the reduction of false alarm. The proposed framework is presented in Figure 2. The proposed framework is a combination of three tiers.

- **Cloud-tier:** The cloud landscape is proficient to perform large computations and data processing tasks for IDS. The computational task is handled by this layer. The dew-tier overcomes the issue of communication overhead in cloud-tier for uploading of the data, the data pre-processed at dew-tier and processed information is uploaded on the cloud servers.

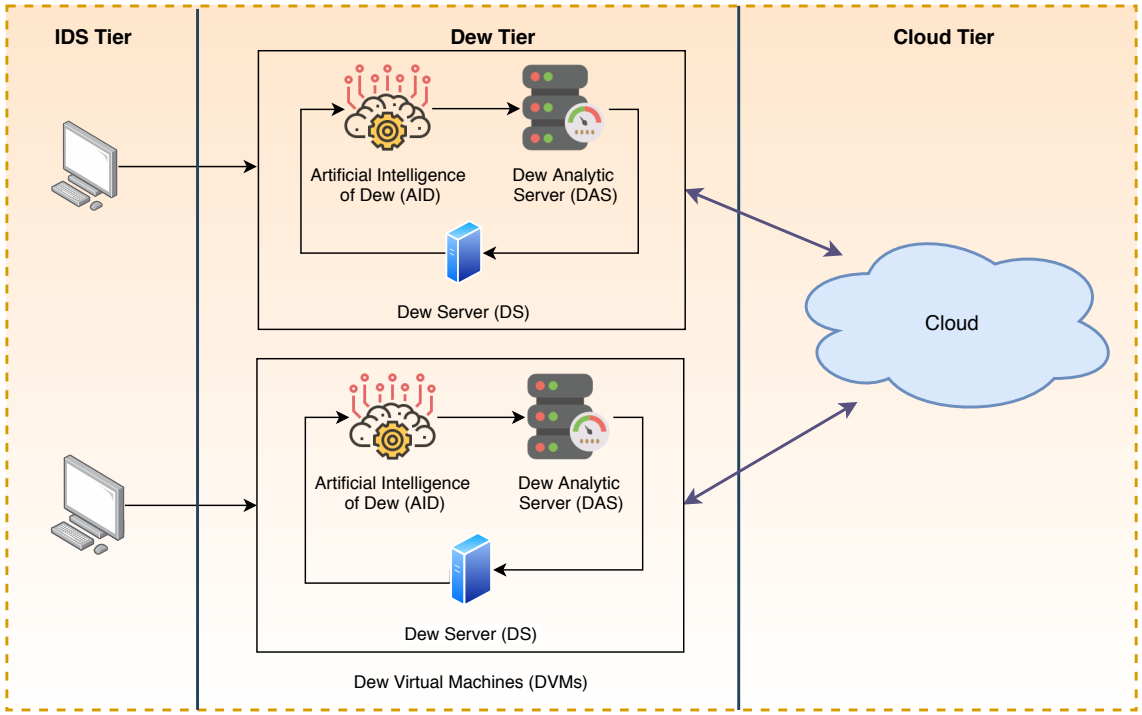

Fig. 2. The proposed framework for Dew Computing as a Service for Intelligent Intrusion Detection

- **Dew-tier:** This tier is equipped with the operating system, utility software, and application software. The algorithm selection is performed on this layer.They enhance the local decision-making process with low latency and fast processing for the FPR minimization process. It also stores the recent data from the traffic and further uploads the data on Cloud after a certain interval decided by the user.
- **IDS-tier:** The intrusion detection is performed at this tier. Numerous detection nodes shared the data to perform the detection task. The adaptive algorithm selection method performed at this layer to operate either on the dew computing tier or cloud tier.

The interaction of the cloud environment, DaaS Eco System and IDS nodes are described in Algorithm 1. Once the connection has been built among all the three tiers. The IDS nodes dispatch the data to edge devices. These edge devices further perform the processing of data. The data received from all the edge devices are sent to the Dew server.

Edge Manager (EM) handles all the communication with edge devices for standardization of data, DBN-based Intrusion Detection, Control System, and False alarm reduction. The process of interaction is described in Algorithm 1. Step 1, it used to fetch the features from the data and prepare the standard alarms list. The selection of DaaS or Cloud services is decided in Step 2, where the DBN-based IDS is to be executed. This decision is taken based on the number of Dew-servers concerning edge devices. EM then broadcast the results to the control system. In step 3, the result received from DBN-based IDS is verified against false alarms. The true alarms are updated with a cloud server for advanced investigation and IDS nodes for intrusion detection.

Algorithm 1 IDS-tier and Dew-tier Interaction in DaaS

1: **Standardization of Data**
   1) Dew Server (DS) received the Data (DT)
   2) Standardization process extract the features from DT
   3) DT represents with Standard Alarms (SDA)
2: **DBN-based Intrusion Detection System**
   1) DS server send the SDA to DBN-based IDS System
   2) DBN-based IDS analyzed the SDA
3: **Control System**
   1) Control System (CS) received the DBN-based IDS results
   2) CS analyze the results and decide the action
4: **Detection of False Alarms**
   1) DaaS received the DBN-based IDS system results and SA
   2) Reduce the False Alarms
   3) True Alarms (TA) is the output
   4) Send TA to the Edge manager
5: **Edge IDS Nodes updated with the True Alarms**

C. DBN-based Intelligent Intrusion Detection

1) Deep Belief Network (DBN): DBN is one of the renowned classifiers for the neural system. RBM is an unaided learning system used by multilayers. In DBN, each layer units are free, as per the unit estimation in the above layer. The DBN model is represented in Figure 3. There are obvious and shrouded layers in DBN. There are two preparation stages. In the primary stage, Contrastive Difference (CD) calculation is used for preparing RBM layers [27]. In the next stage,

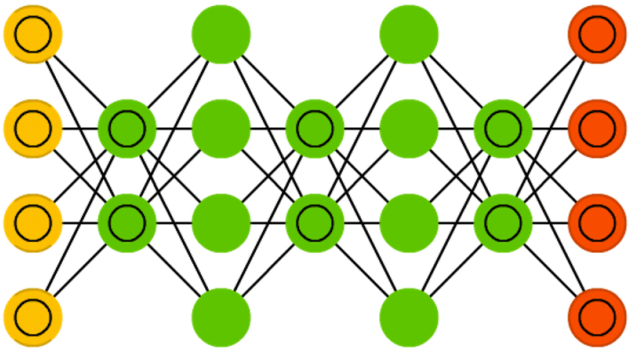

Fig. 3. DBN Model [19]

adjustment of complete DBN parameters is made. At RBM top level, there are heaps learned in an indirect relationship by dispersal fitting in the penultimate layer.

The main task in the DBN is to select the DBN structure which gives the highest accuracy in the system. To obtain the desired accuracy, we performed the classification in three steps. In the first step, the training dataset and the testing dataset is prepared by collecting the incoming traffic. The intrusion features are extracted in the second step. In the last, classification is performed using DBN as shown in Figure 4.

2) Revamped DBN Model: In this paper, we presented the Dew computing framework and revamped DBN model to enhance the classification accuracy. The DBN model is improved by incorporating the likelihood function in the RBM model for unsupervised training. RBM model is trained using Contrastive Divergence (CD) Algorithm [28], the update equation for the parameters can be calculated as per Eq. 1.

$$\begin{aligned} \frac{dl(\theta)}{d\alpha_i} &= (vs_i)_{dt} - (vs_i)_{rc} \\ \frac{dl(\theta)}{d\beta_j} &= (hd_j)_{dt} - (hd_i)_{rc} \\ \frac{dl(\theta)}{dw_{ij}} &= (vs_i hd_j)_{dt} - (vs_i hd_j)_{rc} \end{aligned} \tag{1}$$

The expectation from the training dataset is represented with $(.)_{dt}$, and distribution expected is defined with reconstruction model $(.)_{rc}$. Gibbs sampling [29] is used by CD algorithm to obtain the good fit. The Gibbs steps are used in the CD algorithm recursively for calculation speed improvement, numerous regular updates while ensuring the accuracy of the calculation.

3) Restricted Boltzmann Machine (RBM): RBM is the vital component in deep belief networks and plays an important role in pattern recognition, data classification and data reconstruction. RBM has the hidden layer and visible layers from an undirected graph model. The layers are not connected at the same level, but they are connected with different layers as shown in Figure 3. RBM is used by the neural network for the first time as an energy model. The hidden neurons are represented by hd, and vs is used to represent the visible

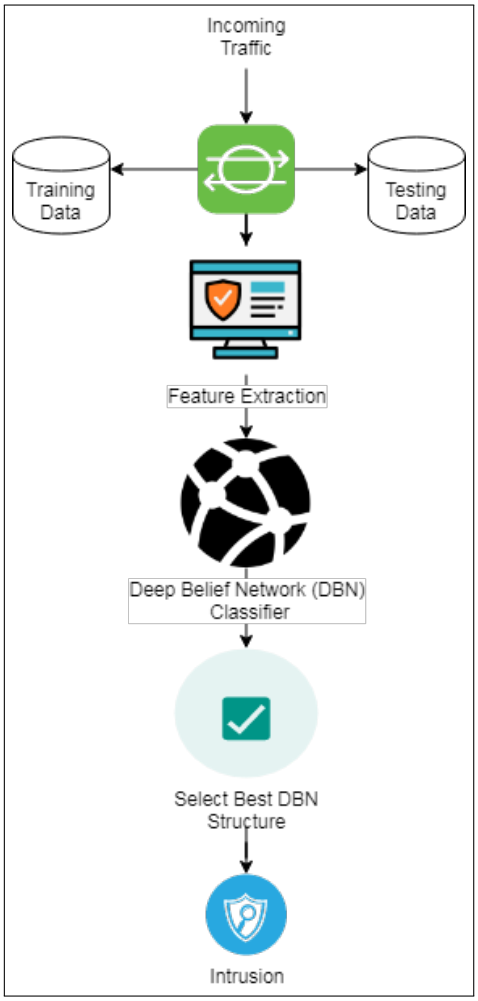

Fig. 4. DBN-based IDS for Edge Ecosystem

layers. The RBM's energy EG function is represented in Eq. 2.

$$EG(vs, hd|\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} vs_i w_{ij} hd_j - \sum_{j=1}^{J} \alpha_i vs_i - \sum_{j=1}^{J} \beta_j hd_j \tag{2}$$

The biases of the visible unit represents with α and hidden unit biases depicted with $beta$. The weight of the connection between vs_i and hd_i represents with w_{ij}, where RBM θ parameters are $w_{ij}, \alpha_i, \; and \; \beta_j$. Bernoulli distribution is used in hidden and visible units. RBM's marginal difference MD is shown in Eq. 3.

$$MD(vs|\theta) = \frac{\sum_{j=1}^{J} e^{-EG(vs,hd|\theta)}}{\sum_{vs}\sum_{hd} e^{-EF(vs,hd|\theta)}} \tag{3}$$

In the training process, the current sample i is the subset of total samples I. The maximum likelihood function is used to find the θ parameter as per Eq. 4.

$$\theta = MAX_{w_{ij},\alpha_i,\beta_j}(\frac{1}{I}\sum_{i=1}^{I} \; in \; MD(vs^i|\theta)) \tag{4}$$

The stochastic gradient ascent is used to obtain the optimal value of θ in RBM.

4) Sparsity in RBM Model: Machine Learning (ML) has a fundamental concept of sparsity and is used in many ML algorithms. When a few coefficients of the model are non-zero, then machine learning algorithms give the sparse result. This property helps in computational advancement, robustness in statistics and fast optimization. In this research, we extend the RBM model to work with a small number of hidden layers and node activation is increased through high sparseness. Sparsity penalty can be persuaded in the hidden layer's sparse state. The probabilities of all hidden units can be limited to represent the low-level features in a useful way. The training with sparse constraints helps to escalate the classification accuracy and reduce the training overfitting risk.

The training samples $TS = vs_1, vs_2, ...vs_n$, the pre-training is performed using unsupervised sparse for model optimization. The sparse penalty is added into the model and resultant objective function is defined in Eq. 5.

$$MAX_{w_ij,\alpha_i,\beta_j}(l(\theta) \ + \ C_1 \sum_{j=1}^{J}(DG(sc||\hat{sc_j}) \ + \ e^{\frac{sc_j^2}{2\delta^2}})) \quad (5)$$

Kullback-Leibler Divergence DG is calculated as per Eq. 6.

$$DG(sc||\hat{sc_j}) = sc \ log \ \frac{sc}{\hat{sc}} \ + \ (1-sc) \ log \frac{1-sc}{1-sc_j} \quad (6)$$

Non-mean Gaussian distribution function is denoted with $e^{\frac{sc_j^2}{2\delta^2}}$. The activation degree is limited with regularization parameter C_1 for the neurons of the hidden layer, this helps to implement sparsity in neurons. The sparsity parameter defined as sc. If $sc = \hat{sc_j}$, we achieve the minimum KL divergence, where $\hat{sc_j}$ represents the average value of activation in j hidden unit. Probability of activation denoted with sc_j and variance factor represented with δ.

The activation rate is mainly focused on this work to keep the neuron activation at the minimum level. This is achieved with a sparsity penalty embedded in training data to avoid the homogeneity features.

5) Normalization of Features: Feature selection is a key in the proposed mechanism, this helps to develop a subset without redundancies. It also boosts the accuracy to a significant level. First, we obtain the numerical values of all the nominal parameters. The min-max scaling is applied to normalization as per Eq. 7.

$$p_{i,j} = \frac{p_{i,j} - min(p_{i,j})}{max(p_{i,j} - min(p_{i,j})} \quad (7)$$

The $p_{i,j}$ defines the parameter at column j and row i of the data frame. RBM layers are used to generate the model.

6) Training of DBN: The collected parameters of the dataset from the incoming traffic are given to DBN after normalization. We start with a low number of hidden layers. To update the weight (wt), then Eq. 8 is used where the learning rate σ is greater than 0.

$$wt_i = \sigma(< \ vs_i hd_j \ > data - < vs_i hd_j \ > \ connection) \quad (8)$$

Hidden units are described with (hd) and (vs) is used to define the visible units. Energy (EN) can be calculated as per Eq. 9.

$$EN(vs, hd) = -\sum_{ij} b_j hd_j wt_{ij} - \sum_{j} b_j h_j - \sum_{i} a_i vs_i \quad (9)$$

We keep on retraining the DBN IDS with a different number of hidden units. First, we start with a few layers and keep on increasing. It takes several trials to find the best DBN structures with good accuracy.

IV. Evaluation

A. Description of UNSW-NB15 Dataset

UNSW-NB15 [30] is a dataset containing assaults that are synthetic contemporary and normal activities. Tcpdump instrument is used for making a bundle of the crude system of the dataset including 49 features. The class marks are created by 12 algorithms, Bro-IDS and Argus. There are a total of 25,400,443 records. The complete dataset segments are divided into training and test sets using various levelling methods. The dataset for preparation consists of 175,341 records. There are 82,332 records in the testing dataset. The information collection apportioned collection consists of 43 features, with 6 highlights (i.e. sport, dstip, Stime, and Ltime) from the complete dataset. There are ten classifications in the divided dataset. It consists of nine assaults and one typical: worms, shellcode, secondary passage, examination, observation, DoS, fuzzes, misuse and nonexclusive. Figure 5 depicts the New training records for balancing the UNSW-NB15 dataset.

The oversampling is required in the UNSW-NB15 dataset due to unbalancing issues in the training samples. The training dataset is balanced using an oversampling method. The new records are generated to balance the training dataset. Figure 5 shows balanced training dataset by applying an oversampling approach.

B. Performance Evaluation of Proposed DBN-based IDS

As depicted in Table I and Figure 6, as there is an increase in hidden units, accuracy also increases. But after some instances, accuracy decreases with an increasing number of hidden units.

The proposed DBN model is trained with a new training dataset that is balanced. Firstly, for DBN layers 1 and 2, 10 hidden units are considered. Slowly the hidden layer numbers are increased to 900. The hidden unit numbers are changed in layer 1 and 2. for performance variations. Table I depicts the experimental results. The best accuracy is given when layer one is at 66 and layer 2 at 60. The confusion matrix is depicted in Table II.

In Table I, as we see there are three classes in hidden layers structure. The first class of structure contains the same number

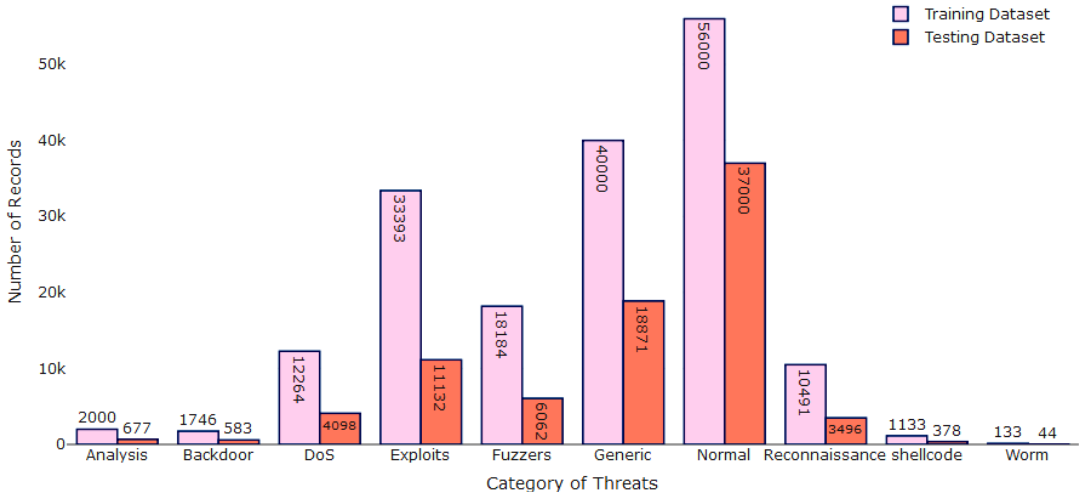

Fig. 5. New training records for balancing the UNSW-NB15 dataset

TABLE I
DBN STRUCTURES AND THEIR CORRESPONDING ACCURACY

Serial No.	No. of Hidden Units (Layer 1)	No. of Hidden Units (Layer 2)	Accuracy (%)
3	15	15	82.15
4	20	20	73.3
5	25	25	80.55
6	27	27	83.62
7	33	33	83.72
8	35	35	82.23
9	40	40	82.52
10	45	45	83.75
11	45	40	83.52
12	50	45	83.32
13	50	55	84.35
14	60	65	85.59
15	65	60	85.7
16	66	60	85.71
17	70	65	84.55
18	200	200	5.5
19	450	200	2.13
20	650	230	0.68
21	700	250	2.33
22	900	240	1.22

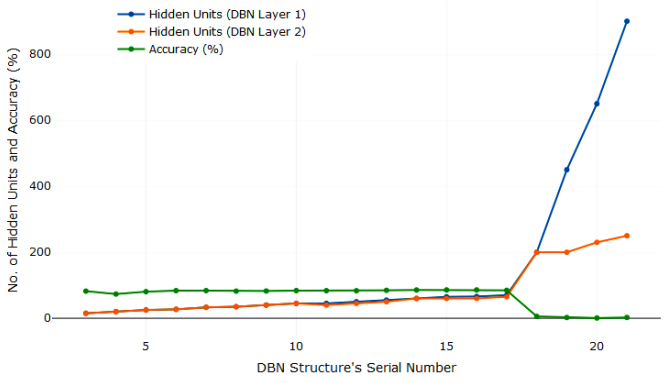

Fig. 6. DBN structure and corresponding accuracy

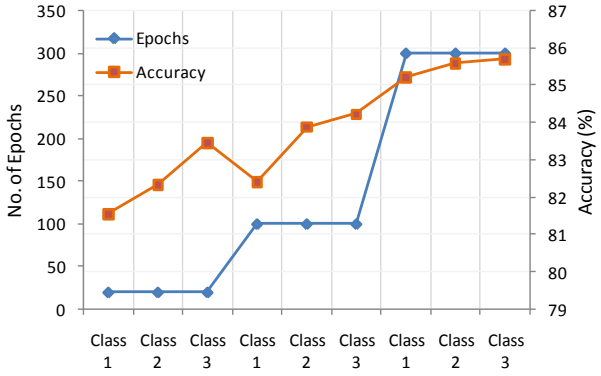

Fig. 7. Accuracy variation with different number of epochs for DBN Structure Classes

of hidden units in layer 1 and layer 2. A case in point, L1(25)-L2(25) provides 80.55% accuracy. The second class has more number of hidden units in layer 2 than hidden layer 1. A case in point, L1(60)-L2(65) gives 85.59% accuracy. In the third class, there are more number of hidden units in layer 1 as compared to layer 2. A case in point, L1(66)-L2(60) achieves the accuracy level up to 85.71%. Furthermore, we evaluated the proposed DBN model by varying the number of epochs to analyze the effect on accuracy. Figure 7 shows the performance of the proposed DBN IDS model by considering the different number of epochs. The experiment result demonstrates that the epoch put a high impact on the accuracy of the DBN model. The accuracy of the model elevates with increasing the number of epochs. Besides, the hidden layers can be increased to a certain level only to achieve better accuracy in the DBN model.

The detection performance of the proposed DBN-based IDS is compared for all the categories of the UNSW-NB15 dataset with SVM and ANN models. The trained DBN weight matrix is used for the ANN initial weight. The pattern of the performance of DBN-IDS and ANN is quite similar in many

TABLE II
CONFUSION MATRIX

	Worm	Shellcode	Reconnaissance	Normal	Generic	Fuzzers	Exploits	DoS	Backdoor	Analysis
Worm	0	0	1	1	0	6	36	0	0	0
Shellcode	0	0	189	12	0	82	95	0	0	0
Reconnaissance	0	0	2560	98	28	195	604	11	0	0
Normal	0	0	1468	25943	8	7800	1722	59	0	0
Generic	0	0	48	14	18168	176	459	6	0	0
Fuzzers	0	0	341	1172	5	3471	1011	62	0	0
Exploits	0	0	378	96	23	863	9682	90	0	0
DoS	0	0	138	65	32	338	3455	61	0	0
Backdoor	0	0	28	10	3	168	343	31	0	0
Analysis	0	0	1	12	3	158	472	31	0	0

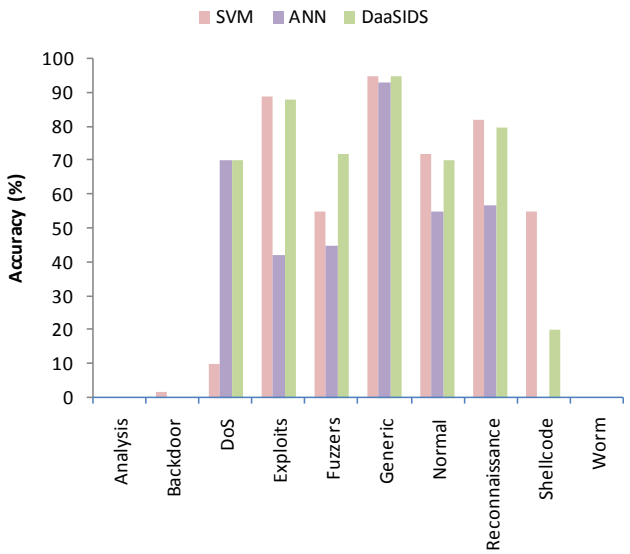

Fig. 8. Performance of SVM, ANN and Proposed System for UNSW-NB15 dataset

of the cases. In ANN, we used the backpropagation method of IDS. The result of the performance in terms of accuracy is shown in Figure 8. The standard approach is followed for the SVM. As we see in Figure 8, the performance of the proposed intrusion detection system is better under DoS, Exploits, Fuzzers, Generic and Normal category. In general, we get the accuracy up to 95%. In the threat category of Analysis, Backdoor and Worm, the IDS gets lower accuracy.

C. Performance Evaluation of Proposed Framework

Total 20 snort nodes are deployed in the IT network to evaluate the performance of proposed and existing frameworks. The set of 8 features (destination port number, destination IP address, source port number, source IP address, packet type, priority, classification, and description) can be extracted from the snort alarms. Similar experiment conducted by Wang et al. [31] (ML-EdgeIDS) and Meng et al. [32] (EdgeIDS) to analyze the difference in workload and delay.

To evaluate the performance of the proposed DaaS approach, we performed the comparison with two most relevant

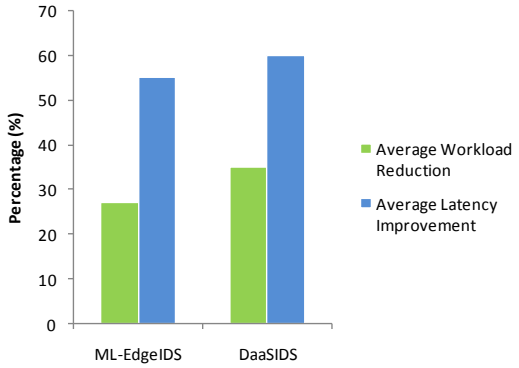

Fig. 9. Workload reduction and latency improvement of ML-EdgeIDS [31] and DaaSIDS in comparison with EdgeIDS [32]

studies [32] and [31]. Meng et al. [32] used the edge environment for intelligent false alarm filter, while Wang et al. [31] enhanced this study by using machine learning in edge computing.

The workload reduction and latency improvement comparison is shown in Figure 9 based on data and network communication. The experiment results show that the proposed DaaS IDS framework is able to reduce the workload of cloud servers up to 35% and improve the latency up to 60% as compared to EdgeIDS [32]. Further, we compared the DaaS IDS with ML-EdgeIDS [31], where we find the 7% reduction of workload and 5% improvement in latency with similar computation capacity. The reason for this improvement is the stability in the resources, while edge computing devices join the network on the ad hoc basis. Network bandwidth is another main concern in edge computing. It has been evident from the experimental evaluation that the proposed DaaS framework is able to improve the false alarm rate with DBN along with improvement in workload reduction and latency in the cloud ecosystem.

V. CONCLUSION

The development of promising false alarm filters for intrusion detection systems (IDS) is a challenging task. Cloud computing enables the offloading of high computation tasks on the cloud helps to design the complex IDS. However, this raises the communication delay and additional cost. In the Edge ecosystem, there is an exchange of a large amount of

data. In this paper, the proposed DaaS for IDS acts as a middleware between the cloud and edge devices. It provides real-time intrusion detection by minimizing the false alarm ratio. Deep Belief Network (DBN) is applied for the classification of attacks. The edge devices used the developed model at the dew computing layer. The epochs are varied to find the best DBN structure. The proposed method is compared with ANN and SVM techniques. The public dataset for intrusion detection known as the UNSW-NB15 dataset is used to compare the proposed model. The experiment result demonstrates that the proposed DaaS for intrusion detection in the Edge ecosystem gives better accuracy with minimal communication delay. Furthermore, the DaaSIDS is able to reduce the workload of cloud servers up to 35%, and improves the latency up to 60% as compare to EdgeIDS. In future, the integration of Edge, Dew, Fog and Cloud ecosystems can be designed to reduce the latency of the overall network. Adaptive IDS can be developed to further improve the classification accuracy.

REFERENCES

[1] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[2] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, "Edge of things: the big picture on the integration of edge, iot and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2017.

[3] P. Singh, S. Krishnamoorthy, A. Nayyar, A. K. Luhach, and A. Kaur, "Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system," *International Journal of Distributed Sensor Networks*, vol. 15, no. 10, p. 1550147719883132, 2019.

[4] Y. Meng *et al.*, "Adaptive false alarm filter using machine learning in intrusion detection," in *Practical applications of intelligent systems*. Springer, 2011, pp. 573–584.

[5] A. Saleem, A. Khan, S. U. R. Malik, H. Pervaiz, H. Malik, M. Alam, and A. Jindal, "Fesda: Fog-enabled secure data aggregation in smart grid iot network," *IEEE Internet of Things Journal*, 2019.

[6] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni, "A trust-aware, p2p-based overlay for intrusion detection," in *17th International Workshop on Database and Expert Systems Applications (DEXA'06)*. IEEE, 2006, pp. 692–697.

[7] W. Meng, W. Li, Y. Xiang, and K.-K. R. Choo, "A bayesian inference-based detection mechanism to defend medical smartphone networks against insider attacks," *Journal of Network and Computer Applications*, vol. 78, pp. 162–169, 2017.

[8] C. J. Fung, J. Zhang, I. Aib, and R. Boutaba, "Robust and scalable trust management for collaborative intrusion detection," in *2009 IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2009, pp. 33–40.

[9] W. Li, W. Meng *et al.*, "Design of intrusion sensitivity-based trust management model for collaborative intrusion detection networks," in *IFIP International Conference on Trust Management*. Springer, 2014, pp. 61–76.

[10] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine-learning-based android malware detection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018.

[11] W. Li, W. Meng *et al.*, "Sooa: exploring special on-off attacks on challenge-based collaborative intrusion detection networks," in *International Conference on Green, Pervasive, and Cloud Computing*. Springer, 2017, pp. 402–415.

[12] W. Li, W. Meng, H. H. S. Ip *et al.*, "Pmfa: toward passive message fingerprint attacks on challenge-based collaborative intrusion detection networks," in *International Conference on Network and System Security*. Springer, 2016, pp. 433–449.

[13] Q. Liu, G. Wang, X. Liu, T. Peng, and J. Wu, "Achieving reliable and secure services in cloud computing environments," *Computers & Electrical Engineering*, vol. 59, pp. 153–164, 2017.

[14] A. Jindal, A. Schaeffer-Filho, A. K. Marnerides, P. Smith, A. Mauthe, and L. Granville, "Tackling energy theft in smart grids through data-driven analysis," in *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 410–414.

[15] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection for grid and cloud computing," *It Professional*, vol. 12, no. 4, pp. 38–43, 2009.

[16] W. Yassin, N. I. Udzir, Z. Muda, A. Abdullah, and M. T. Abdullah, "A cloud-based intrusion detection service framework," in *Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*. IEEE, 2012, pp. 213–218.

[17] T. Alharkan and P. Martin, "Idsaas: Intrusion detection system as a service in public clouds," in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE, 2012, pp. 686–687.

[18] Y. Ding, S. Chen, and J. Xu, "Application of deep belief networks for opcode based malware detection," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3901–3908.

[19] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1. IEEE, 2017, pp. 639–642.

[20] J. Kaiser, D. Zimmerer, J. C. V. Tieck, S. Ulbrich, A. Roennau, and R. Dillmann, "Spiking convolutional deep belief networks," in *International Conference on Artificial Neural Networks*. Springer, 2017, pp. 3–11.

[21] J. Koo and D. Klabjan, "Improved classification based on deep belief networks," *arXiv preprint arXiv:1804.09812*, 2018.

[22] Q. Tian, D. Han, K.-C. Li, X. Liu, L. Duan, and A. Castiglione, "An intrusion detection approach based on improved deep belief network," *APPLIED INTELLIGENCE*, 2020.

[23] Y. Meng, W. Li, and L.-F. Kwok, "Towards adaptive false alarm reduction using cloud as a service," in *2013 8th International Conference on Communications and Networking in China (CHINACOM)*. IEEE, 2013, pp. 420–425.

[24] Y. Wang, "Definition and categorization of dew computing," *Open Journal of Cloud Computing (OJCC)*, vol. 3, no. 1, pp. 1–7, 2016.

[25] K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," *Open Journal of Cloud Computing (OJCC)*, vol. 2, no. 1, pp. 16–24, 2015.

[26] S. Ristov, K. Cvetkov, and M. Gusev, "Implementation of a horizontal scalable balancer for dew computing services," *Scalable Computing: Practice and Experience*, vol. 17, no. 2, pp. 79–90, 2016.

[27] A. A. Ramaki, A. Rasoolzadegan, and A. G. Bafghi, "A systematic mapping study on intrusion alert analysis in intrusion detection systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–41, 2018.

[28] E. R. Merino, F. M. Castrillejo, and J. D. Pin, "Neighborhood-based stopping criterion for contrastive divergence," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 2695–2704, 2017.

[29] M. Fatemi, K. Granström, L. Svensson, F. J. Ruiz, and L. Hammarstrand, "Poisson multi-bernoulli mapping using gibbs sampling," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2814–2827, 2017.

[30] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[31] Y. Wang, W. Meng, W. Li, Z. Liu, Y. Liu, and H. Xue, "Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 19, p. e5101, 2019.

[32] W. Meng, Y. Wang, W. Li, Z. Liu, J. Li, and C. W. Probst, "Enhancing intelligent alarm reduction for distributed intrusion detection systems via edge computing," in *Australasian Conference on Information Security and Privacy*. Springer, 2018, pp. 759–767.